

分散エージェントを利用したソフトウェア構築手法の提案 ——データベースアプリケーション構築への適用

井上 晃[†] 阿部 徹 治^{†,☆}
満 永 豊[†] 加藤 康 之^{†,☆}

現在、様々なオブジェクト指向ソフトウェア開発環境が提案されている。しかしながら、たとえば企業内のデータベースシステム等の開発に適用した場合、対象領域やデータベースに関する専門的な知識が必要になるという課題が残る。さらに、その保守性に問題のあるものも多い。本論文では、この問題を解決するため、ネットワーク上に分散するエージェントが協調して、アプリケーションを構築する手法を提案する。この手法では、ユーザは抽象的なアプリケーション作成仕様を指定する。すると、エージェントは協調による問題解決技術の利用により、作成仕様を解釈し、アプリケーションを構成する複数の部品オブジェクトを自動生成する。そして、それらを組み合わせ、目的のアプリケーションを構築する。これにより、ユーザに要求される知識量を軽減し、保守性を改善する。また、この提案手法をデータベース検索や統計処理を行うデータベースアプリケーションの構築という問題に適用し、エージェントの機能や知識表現方法、分散エージェントの協調方法を具体的に示す。そして、現実の社内システムで利用可能なデータベースアプリケーションを構築するプロトタイプシステムを作成し、動作の検証と実問題における本手法の有効性を示した。

Software Construction Method Using Distributed Agents —Employed for the Construction of a Database Application

AKIRA INOUE,[†] TETSUJI ABE,^{†,☆} YUTAKA MITSUNAGA[†]
and YASUYUKI KATO^{†,☆}

Recently, several object-oriented software development environments have been proposed. However generally these environments present some problems: for example, they require expertise regarding application construction, moreover neither the construction process nor maintenance are easy. So we propose a new software construction method using distributed agents. The construction environment obtained using this method is flexible and maintenance efficiency is improved. A user orders this environment to make an application by specifying the output image of the application. The agents cooperatively understand the specification and produce some part-objects which are components of the application. They then automatically construct the application from these part-objects. Furthermore, by applying this method to an actual problem such as the construction of a database application, we provide concrete examples of the agents' features, which include their functions, ways of expressing knowledge, and cooperation procedures. The database application can retrieve data from the database or perform statistical processes. Moreover by building a prototype system to produce a database application, which can be used in an actual corporate system, we confirm the operation and effectiveness of this method for solving actual problems.

1. はじめに

現在、ソフトウェア構築環境では、オブジェクト指向を用いた設計方法や開発手法が注目されつつある^{1),2)}。その中で、我々は、システムコンセプトからコーディ

ングに至る全ステージで完全なオブジェクト指向アプローチを貫いた統合オブジェクト指向技術を開発し³⁾、社内のデータベース検索・統計処理等のデータベースアプリケーション開発に適用している(図1)。

この開発環境では、ユーザはコーディングをまったく行わずに、アプリケーションを構成するソフトウェア部品(以下、部品オブジェクトと呼ぶ)を自由自在に構築できる。そして、この部品オブジェクトを組み合わせ、アプリケーションを構築することが可能であ

[†] NTTアクセス網研究所
NTT Access Network Systems Laboratories
[☆] 現在、株式会社サイバーラボ
Presently with Cyber Laboratory Inc.

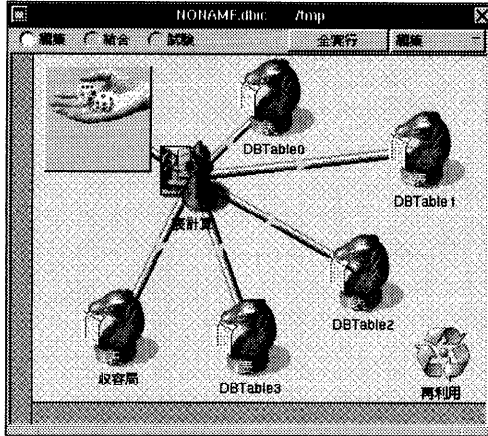


図1 オブジェクト指向技術を用いたソフトウェア構築環境例
Fig. 1 An example of the software construction environment implemented by using integrated object-oriented technology.

る。さらに、アプリケーションや部品オブジェクトに完全な階層化の概念を導入しているため、作成されたアプリケーションを1つの部品オブジェクトと取り扱うことができ、さらに複雑・高性能なアプリケーションを構築できる。このようなオブジェクト指向プログラミング環境を提供することにより、従来と比較して2桁近い開発効率の向上を実現している。しかしながら、この構築環境には次の課題がある。

課題1 専門知識 ユーザが部品オブジェクトを作成し、組み合わせ、アプリケーションを自由に構築できる環境は提供されている。しかしながら、1つのアプリケーションを複数の部品オブジェクトの組合せとして設計するには、対象とする問題領域に関する専門知識や業務分析が不可欠である。さらに、部品オブジェクトの設計には、アプリケーションそのものに関する知識が必要である。たとえば、データベースを検索し、統計処理等を行うデータベースアプリケーションの場合、データベースに関する知識がなければ、部品の設計は困難である。つまり、テーブル構造やその項目、テーブル間の関連等を考慮して、部品を作成する必要がある。部品の組合せにも、データベースアプリケーションが対象とする問題領域に関する専門知識が不可欠である。

課題2 保守性 アプリケーションを自由に作成、変更できる環境を提供しているため、ユーザ自ら、必要なアプリケーションをいろいろな場所で作成し、改造・再利用することが可能である。このようにアプリケーションの拡張性、柔軟性が優れているというメリットはあるが、一方で、汎用の部品オブジェクトに

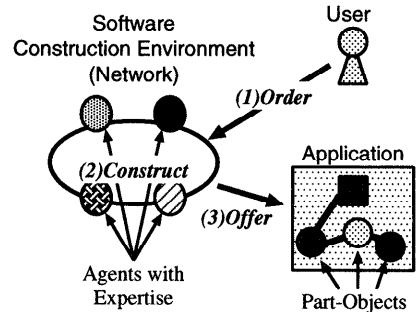


図2 提案するソフトウェア構築手法の概要
Fig. 2 The software construction method proposed in this paper.

問題が生じた場合、その部品を利用しているすべてのアプリケーションに対処しなければならないという保守性の問題がある。同様に、アプリケーションを作成するときに利用した知識等に変更が生じた場合、ユーザは関係するアプリケーションを再度、作り直す必要がある。また、アプリケーションを作成する構築環境自体に問題が生じた場合、ユーザに配布したすべての構築環境に対処する必要があり、さらには、この環境で作成されたアプリケーションも再構築しなければならない。

本論文では、ユーザからの要求に応じたアプリケーションを容易に作成・保守する手法として、分散したエージェント（以下、分散エージェントと呼ぶ）の協調による問題解決技術を用いた、新しいソフトウェア構築手法^{4),5)}を提案し、以上の課題を解決する。

2. 分散エージェントを利用したソフトウェア構築手法の概要

提案するソフトウェア構築手法は、分散エージェントの協調による問題解決技術と部品生成技術により実現される。ここで、エージェントを次のように定義する。「専門知識を有する通信可能アプリケーションの一種であり、外部からの要求を認識し、自立的・自発的に動作を決定する」。さらに専門知識とは、データベース構造に関する知識や統計処理という機能に関する知識、対象問題領域に関する知識など、アプリケーション構築に必要な様々なレベルの知識を指す。本手法によるアプリケーション構築過程の概要を示す(図2)。

(1) ユーザは構築を希望するアプリケーションの仕様を最終的な出力形式で指定し、構築環境に作成を依頼する。ここでいう最終的な出力形式の指定とは、アプリケーションが最終的に出力す

る帳票上のカラム名の指定やグラフの縦軸横軸の指定などである。

- (2) 分散エージェントは、相互協調してユーザからの作成依頼を解釈し、アプリケーションに必要な部品オブジェクトを作成する。
- (3) すべての部品オブジェクトが用意されると、エージェントはそれらを組み合わせ、アプリケーションを作成する。そして、ユーザに提供する。

このように分散エージェントの協調により部品オブジェクトを生成し、アプリケーションを構築する手法は、従来手法と比べ、いくつかの利点がある。それについて、分散エージェントの協調を用いたことによる利点と、部品生成という機構を用いたことによる利点のそれぞれについて示す。まず、分散エージェントの協調による利点は、次の3点である。

- ユーザに必要とされる専門知識が減少する。すなわち、エージェントがそれぞれ専門知識(対象領域や構築に関する知識)を持っているために、ユーザはアプリケーション構築に関する一部の知識を用意すればよい。
- 構築処理の分散化が容易になる。構築環境は分散したエージェントにより構成されるため、各エージェントを複数の計算機上で稼働させることにより、容易に分散処理可能である。
- 構築環境の保守性が向上する。これは、環境が分散エージェントにより構成されるため、構築環境の一部に問題がある場合には、該当エージェントだけを取り替えればよく、対応が容易である。

一方、部品生成機構による利点は、次の2点である。

- 構築環境の拡張性・柔軟性が優れる。エージェントはある範疇ごとに知識を持ち、その範疇に関係するほとんどの部品オブジェクトを作成できる。つまり、あらかじめすべての部品を用意する代わりに、1つのエージェントを用意し、動的に部品を生成させることが可能となる。さらに、各エージェントが協調し、動的に部品を組み合わせることにより、あらかじめすべてのアプリケーションを用意するときに生じる組合せの爆発を防ぐことができ、様々な作成要求に柔軟に対応できる。この点は従来の部品生成系システムの利点の1つであるが、本手法ではさらに、エージェント間協調による、柔軟な部品組合せを実現し、効率的な構築が可能である。
- 作成されるアプリケーションの保守性が向上する。アプリケーションを構成する部品は構築環境上のエージェントが作成する。そのため、アプリケー

ションの保守は、エージェントの保守に同等である。すなわち、エージェントを保守し、アプリケーションを再構築させることにより、最新の状態を維持することができ、すべてのアプリケーションの保守が容易になる。

なお、本論文では、以上の特徴を明らかにするため、データベースの検索や統計処理を行うデータベースアプリケーション(以下、DB-APと呼ぶ)に本構築手法を適用する。これは、以下の点において、本手法に適当な問題領域である。

- 一般にDB-APでは、複数のデータベーステーブルに対する様々な検索結果を統計処理する。つまり、複数の検索処理の組合せであり、各処理は基本的に独立である。したがって、各処理を部品オブジェクトに対応させ、それらを組み合わせてアプリケーションを構築することは容易であり、本手法に適当である。
- DB-APの作成には、データベースに関する知識とアプリケーションの対象領域に関する知識が要求され、従来手法では両方の知識を兼ね備えたユーザしか構築できなかった。これに対して、本手法ではエージェントがそれらの知識を持つことにより、構築に必要なユーザの知識量を少なくでき、より多くのユーザが利用することができる。
- DB-APでは、データベース構造等の変更にとまなうアプリケーションの設計変更が多く、拡張性、柔軟性に富む構築手法が必要である。本手法では、データベース構造の変更は、該当エージェント内の知識の変更に等しい。したがって、構築環境の拡張や保守は、エージェントのそれに等しく、柔軟に対応できる。
- データベースが大きくなるに従い、テーブル数や項目数も増え、テーブル間の関連等は指数級数的に複雑になる。そのため、あらかじめそれらを構築環境内に記述することは非現実的であり、すべてのアプリケーションを用意することも難しい。本手法では、範疇別に基本的な知識を用意すればよく、テーブル間の関連等は、エージェントの協調によって動的に生成され、それに基づき、アプリケーションを構築する。したがって、データベース等が複雑になっても対応可能である。

なお、本論文では一適用例として、DB-APを採用したものであり、構築手法の基本概念はDB-APに依存するものではない。たとえば、次の問題領域にも本構築手法を適用可能である。いずれも、レベルの違う複数の専門知識を必要とし、求められる条件等が変化する

る場合には、アプリケーションの再構築が必要となる。

- 設備の更改・新規建設に関する設計支援アプリケーションの構築
- 設備に関する需要予測を行うシミュレーション系アプリケーションの構築

以上のソフトウェア構築手法を実現するにあたり、次の課題がある。

課題 1 エージェントの機能，知識表現方法

課題 2 分散エージェントの協調手法

この課題に対し，以下，3章では，適用領域であるDB-APに関する業務分析を行う。そして，その分析結果に基づき，エージェントの分類，知識表現方法，分散エージェントの協調によるソフトウェア構築手法を述べる。次に，4章では，プロトタイプシステムを実装し，本構築手法の動作を確認する。5章では，本構築手法を実システムに適用した場合の有効性を考察する。

3. 分散エージェントによるデータベースアプリケーション構築手法

分散エージェントが協調して，DB-APを構築する手法を述べる。本論文では，DB-APの処理対象は，NTTの加入者系設備に関する管理データベースとし，次章で述べるプロトタイプシステムでも同様とする。このデータベースは，テーブル数100以上，項目数約4000，テーブル間の関連数約1000という非常に大規模なものであり，従来の構築手法では，前述した様々な問題がある。以下，この管理データベースを対象とする実システムを分析し，その中のDB-APの構築過程の特徴を抽出する。そして，その特徴を利用して設計したエージェントについて，その分類，構造，知識表現方法を示し，2段階からなる分散エージェントの協調による構築手法について述べる。

3.1 データベースアプリケーションの特徴抽出と分散エージェントへの適用

実システムで利用中のDB-APは，文献3)の技術を利用したソフトウェア構築環境で作成される。つまり，複数の部品オブジェクトを組み合わせ，アプリケーションを構築する。ここで，DB-APを構成する部品オブジェクトとは，次のような機能を持つプログラムの一環である。

「1つのデータベーステーブルを対象として，指定された統計処理条件を満たすSQLスクリプトを保持する。そして，それをを用いて検索・統計処理し，結果を出力する機能を有する。」

このシステムでは，100種類以上のDB-APがユー

ザにより構築され，その設計，構築過程を分析すると，利用する知識の程度により，ほとんどが次の3段階から構成される。

- (1) 抽象的概念によるDB-APの全体設計過程。たとえば，集計対象を「架空ケーブルの長さ」とし，集計条件に「建設年度3年ごと」，統計方法に「総和」を指定する。
- (2) 全体設計に基づき，専門的な知識を利用して詳細設計する過程。すなわち，全体設計に含まれる「架空ケーブルの長さ」等の抽象的な表現をデータベース構造に適合する具体的な表現に展開する。たとえば，「架空ケーブルの長さ」という表現は，2つのデータベーステーブル「架空メタルケーブル」，「架空光ケーブル」のカラム「ケーブル長」に相当する。
- (3) 詳細設計から，実際にアプリケーションを作成する過程。つまり，詳細設計を部品オブジェクトが作成可能な仕様に分解し，それに基づいて部品オブジェクトを作成する。そして，それらを組み合わせ，DB-APを構築する。たとえば，データベーステーブルごとに仕様を分割し，部品オブジェクトを作成する。

次に，この3段階にわたる設計・構築過程を，エージェントにより実現する方法を検討する。まず，第1段階の処理は，ユーザが全体仕様を入力する部分であり，1つのエージェントの機能として実現可能である。一方，第2，第3段階の処理は，以下のように，全体の状況に応じて変化する部分と固定的な部分とに分けられる。固定的な部分は，エージェントの内部機能として実現可能であるが，変化する部分は実現が難しい。

- 第2段階
 - － (変化) 全体設計に対応する専門知識を選択する処理
 - － (固定) 専門知識を用いて具体的な詳細設計を作成する処理
- 第3段階
 - － (変化) 詳細設計を，部品オブジェクトが作成可能な仕様に分解する処理
 - － (固定) 仕様に基づき，部品オブジェクトを作成する処理

以上の分析結果より，本構築手法では，次の3種類のエージェントを用意し，各段階の変化する処理部分は，エージェント間の協調により，実現する。

- 第1段階に相当する「インタフェースエージェント」
- 第2段階の固定的処理に相当する「アプリケー

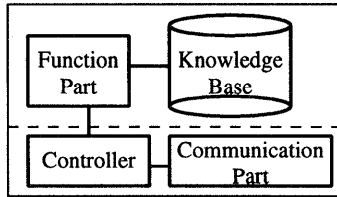


図3 エージェントの基本構造
Fig. 3 Basic structure of an agent.

表1 知識・仕様で用いる表記規則

Table 1 Rules for expressions used in the knowledge base and specifications.

接頭語	意味
(何もつかない)	抽象表現
'A'	文字列 A
Column:A	データベースカラム名 A
Table:A	データベーステーブル名 A
Function:A	演算関数 A

表2 知識・仕様を表現する構造記述子

Table 2 Structural descriptions for the expressions of the knowledge base and specifications.

構造記述子	意味
(A B)	表現 A, または, 表現 B
(A & B)	表現 A, かつ, 表現 B
def(A)={B}	表現 A は表現 B に変換可能
case(A){B}	条件 A が真ならば, 表現 B は真である

ション構築エージェント」

- 第3段階の固定的処理に相当する「部品作成エージェント」

3.2 エージェントの構造, 知識表現

すべてのエージェントに共通な基本構造を図3に示す。基本機能として、エージェント全体の制御を行う部分 (Controller) と、他のエージェントと通信を行う通信機能部分 (Communication Part) がある。さらに、各エージェントに固有の機能部分 (Function Part) と知識ベース (Knowledge Base) がある。ここで、知識ベースにおける知識や、エージェント間で取り交わされる仕様を表現するため、簡易な表記規則を定義する (表1 ~ 表3)。表1は接頭語を用いた表記規則であり、表2はC言語の文法に類似した構造記述子、表3は仕様に関する演算子である。

以下、各エージェントの機能部分と知識ベースについて示す。

3.2.1 インタフェースエージェント

インタフェースエージェント (Interface Agent: I-A) は、構築環境とユーザ間のインタフェース部分に相当し、ユーザから作成仕様を受け付け、他のエージェントにアプリケーション作成を発注する。

表3 仕様に関する演算子

Table 3 Operative functions for specifications.

演算子	意味
$\cup, \cap, -$	仕様の和, 積, 差
$\subset, \supset, \subseteq, \supseteq, \equiv$	仕様の包含関係

$S = \{ \text{検索対象}(), \text{検索条件}(), \text{集計項目}(), \text{集計条件}(), \text{集計方法}() \}$

図4 ユーザが入力する作成仕様 S の形式

Fig. 4 The definition of the specification S received from a user.

構造としては、特別な知識は持たず、次の機能を持つ。

- ユーザから作成仕様を受け付ける機能
- 他のエージェントと協調し、入力された作成仕様を適当なエージェントに依頼する機能
- 作成されたアプリケーションを他のエージェントから受け取り、ユーザに提供する機能

また、ユーザが作成仕様を入力するときには、すべてを自然言語で入力する方式から、選択方式まで様々なものが考えられる。一方、本論文では、対象アプリケーションをDB-APとするため、作成仕様にはデータベースに関する検索、統計処理条件等が必須である。そこで、エージェントによる仕様の解釈処理を軽減するため、作成仕様 S を5項目に分類する (図4)。そして、「集計方法」の項目は選択方式とし、その他の項目には先に定義した表記規則で表現した文字列を入力する。これにより、ユーザが最低限必要とする知識は、表記規則である。

3.2.2 アプリケーション構築エージェント

アプリケーション構築エージェント (Application Construct Agent: AC-A) は、構築対象に関する抽象的な範疇を単位に用意し、その範疇に関する専門知識を有する。たとえば、加入者系設備データベースの場合、次の設備範疇に分けられ、1つのエージェントが複数の範疇の知識を持つことも可能とする。

「架空ケーブル」, 「地下ケーブル」, 「電柱」, 「マンホール」など

機能部分には次の機能がある。

- I-Aから提示される仕様を解釈し、実現可能な部分仕様を抽出する機能
- 部品作成エージェントと協調し、受注した仕様を適当なエージェントに発注する機能
- 作成された部品オブジェクトを組み合わせ、アプリケーションを構築する機能

一方、知識ベースには、図5に示す、構築対象に関

```
def(表現 1|..) = { カラム, テーブル定義リスト 1 }
def(表現 2|..) = { カラム, テーブル定義リスト 2 }
.....
```

図5 アプリケーション構築エージェントにおける知識表現
Fig. 5 The definition of knowledge in the Application Construct Agent.

```
def(ケーブルの長さ|..) = {Column:ケーブル長}
def(ケーブルの建設年|..) = {
    Column:ケーブル建設年}
def(架空ケーブル|..) = {
    Column:ケーブル方式 = ('架空')
    Table = (Table:架空配線, Table:架空光配線,
            Table:ビル配線)}
def(種別|種類) = {
    case(CCP)
        {Column:種別コード = (04-10,29,56,83,84)}
    case(CCP-F)
        {Column:種別コード = (79-82)} }
```

図6 アプリケーション構築エージェントにおける知識表現例
(架空ケーブルの場合)

Fig. 6 A knowledge expression regarding "Overhead Cable" in the Application Construct Agent.

する「抽象表現」(i.e. 設備名)と「データベース内部での具体的な表現」(i.e. データベースカラム名)との対応関係を定義する。たとえば、「架空ケーブル」という設備に関連する知識は、図6のように表現される。

3.2.3 部品作成エージェント

部品作成エージェント (Part Produce Agent: *PP-A*) は、アプリケーションを構成する部品オブジェクトを作成する。前項で述べたように、DB-AP の場合、データベーステーブルを単位とした部品オブジェクトの作成が適当なため、*PP-A* は検索の対象となるデータベーステーブルを単位に用意する。そして、そのテーブルに関する知識を有する。たとえば、加入者系設備の管理データベースの場合、次の各テーブルが単位となり、1つのエージェントが複数テーブルに関する知識を持つことも可能とする。

「架空配線」、「架空光配線」、「ビル配線」など機能部分には、次の機能がある。

- *AC-A* と協調し、提示される仕様の検討、構築可能な部分仕様の抽出を行う機能
- 依頼された仕様に基づき、部品オブジェクトを作成する機能

一方、知識ベースには、対象データベーステーブルに関する3つの情報 (テーブル情報、サーバ情報、統計処理演算関数に関する情報) を定義する (図7)。たとえば、「架空配線」テーブルに関する知識は、図8のようになる。

```
データベーステーブル情報={
    テーブル名 = (Table:テーブル名){
        Column:カラム名 1 = [データ型 1]
        ...
    }
}
```

```
データベースサーバ情報={ ホスト名 = (), DBMS = () }
演算関数情報 = {Function:関数 1 = func1(),...}
```

図7 部品作成エージェントにおける知識表現

Fig. 7 The definition of knowledge in the Part Produce Agent.

```
データベーステーブル情報={
    テーブル名 = (Table:架空配線){
        Column:ケーブル方式 = [String]
        Column:ケーブル長 = [Double]
        Column:種別コード = [Integer]
        ...
    }
}
```

```
データベースサーバ情報={
    ホスト名 = ('database.aaa.bb.jp')
    DBMS = ('SYBASE') }
演算関数情報 = {
    Function:総和 = sum()
    Function:最大値 = max() }
```

図8 部品作成エージェントにおける知識表現例
(架空配線テーブルの場合)

Fig. 8 A knowledge expression regarding the "Overhead Wiring" table in the Part Produce Agent.

3.3 分散エージェントの協調による構築手法

前述した3種類のエージェントの協調によるアプリケーション構築手法を述べる。構築は、2階層から構成される。第1階層では、*I-A* がユーザから作成仕様を受け付け、そして、*AC-A* と協調し、適当な *AC-A* にアプリケーション作成を依頼する。第2階層では、その仕様を *AC-A* が *PP-A* との協調により複数の副仕様に分割し、必要な部品オブジェクトを作成する。そして、組み合わせてユーザにアプリケーションを提供する。

また、エージェント間協調手法には、契約ネットプロトコル方式^{6)~9)}を利用する。しかしながら、従来手法と異なり、仕様を提示されたエージェントは、入札値としてそのエージェントが実現可能な部分仕様を逆提示するという手法をとり、効率的な協調過程を実現する。

以下、各階層の協調過程を (A), (B) に示し、最後に協調過程全体を通した、エージェント間メッセージフローを (C) に示す。

(A) 第1階層

I-A はユーザが入力した作成仕様について、複数の *AC-A* と協調し、その実現可能性を判断する。そして、最適な *AC-A* に作成を依頼する。過程は、STEP1-1 から STEP1-7 の7ステップから構成される。以下、複数

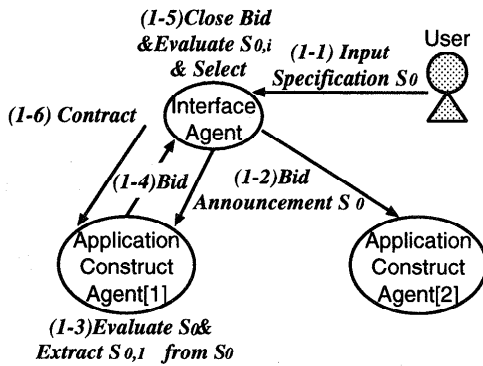


図9 第1階層の協調手順
Fig. 9 Cooperation procedure in first layer.

$S_0 = \{$
 検索対象 = (架空ケーブル, 地下ケーブル)
 検索条件 = (種別 = (CCP | CCP-F))
 集計項目 = (ケーブルの長さ)
 集計条件 = (ケーブルの建設年 on (1991~1995))
 集計方法 = (Function:総和)
 $\}$

図10 0次仕様: S_0
Fig. 10 0th specification: S_0 .

の $AC-A$ を $AC-A_i$ ($i = 1, \dots, M_{AC-A}$) と表す。 M_{AC-A} は、存在する $AC-A$ の総数である。

STEP1-1 はじめに、ユーザは作成仕様を $I-A$ に入力する (図9(1-1))。これを0次仕様 S_0 と呼ぶ。たとえば、「架空ケーブル」と「地下ケーブル」に関する統計処理を行う $DB-AP$ の作成仕様は、図10のようになる。なお、前述したように集計方法は選択方式、それ以外は自由入力である。

STEP1-2 $I-A$ は、入力された0次仕様 S_0 をすべての $AC-A_i$ に提示し、入札を求める (図9(1-2))。

STEP1-3 $AC-A_i$ は、提示された仕様 S_0 を分析し、解釈可能な部分仕様 $S_{0,i}$ を作成する (図9(1-3))。つまり、部分仕様とは、提示仕様からそのエージェントが実現可能な部分を抽出した仕様である。したがって、部分仕様 $S_{0,i}$ と仕様 S_0 との間には、 $S_{0,i} \subseteq S_0$ の関係が成り立つ。提示仕様の実現可能性は次の2段階で判断する。

- (1) 「検索対象」項目に関する知識の有無を確認する。一部でも知識がある場合、次の判断に移る。まったく保持してない場合、実現不可能である。
- (2) 「検索対象」以外の項目について知識の存在を確認する。すべて存在する場合、実現可能とし、逆に、1つでも存在しない場合には、不可能である。

以上の2段階を通して実現可能と判断した場合、実

$S_{0,0} = \{$
 検索対象 = (架空ケーブル)
 検索条件 = (種別 = (CCP | CCP-F))
 集計項目 = (ケーブルの長さ)
 集計条件 = (ケーブルの建設年 on (1991~1995))
 集計方法 = (Function:総和)
 $\}$

図11 部分仕様の例: $S_{0,0}$
Fig. 11 An example of extracted specification: $S_{0,0}$.

際に部分仕様を抽出する。逆に、不可能な場合には、すべての項目が空な部分仕様を作成する。たとえば、図6の知識を持つ $AC-A_0$ に、図10の仕様を提示した場合には、図11に示す部分仕様 $S_{0,0}$ を作成する。

STEP1-4 $AC-A_i$ は、作成した部分仕様 $S_{0,i}$ と $AC-A_i$ が稼働するホストの平均負荷値 LA_i を入札値として、 $I-A$ に逆提案する (図9(1-4))。なお、平均負荷値とは $DB-AP$ 作成に対する $AC-A_i$ の処理困難性を表し、低いほど処理能力に余裕がある。

STEP1-5 $I-A$ は、公示から一定時間 T_a で入札を締め切る。そして、入札結果 ($S_{0,i}, LA_i$) ($i = 1, \dots, M_{AC-A,0}$) を評価し、仕様 S_0 を満足する最適な部分仕様 $S_{0,i}$ の組合せを選択する (図9(1-5))。 $M_{AC-A,0}$ は仕様 S_0 に入札してきた $AC-A_i$ の総数を表す。入札結果の評価・選択方針は、構築されるアプリケーションの品質を重視し、次いで構築の迅速性とする。品質の評価には様々な項目があるが、本論文では、提示仕様に対する $AC-A_i$ の実現能力の高低により評価する。なぜならば、仕様実現能力が高いという事は、その仕様に適合した $AC-A_i$ であることを表し、作成されるアプリケーションの品質が優れると推測できるからである。仕様実現能力は、提示仕様 S_0 に対して逆提案された部分仕様 $S_{0,i}$ の仕様実現率により計測する。つまり、提示仕様 S_0 を広範囲に満足する $S_{0,i}$ ほど、高く評価する。以下、この方針に基づく評価選択アルゴリズムを示す。

- (1) はじめに、式(1)により仕様 S_0 の実現可能性を判断する。つまり、すべての部分仕様 $S_{0,i}$ を組み合わせ、 S_0 を実現可能か否かを判断する。

$$\bigcup_{i=1}^{M_{AC-A,0}} S_{0,i} \equiv S_0 \quad (1)$$

実現可能ならば、次項に進む。逆に、不可能な場合、その仕様は構築不可能であり、ユーザに通知する。

- (2) 次に、部分仕様 $S_{0,i}$ の組合せを考え、 S_0 を満足し、かつ、その組合せ数が最小なものを

選択する。まず、 $S_{0,i}$ の最小組合せ数 $N_{1,\min}$ を次式 (2) により求める。

$$N_{1,\min} = \min_{1 \leq N_1 \leq M_{AC-A,a}} N_1 \quad (2)$$

$$\text{where } \bigcup_{i=1}^{N_1} S_{0,k_i} \equiv S_0$$

$$\left(\begin{array}{l} k_i = 1, \dots, M_{AC-A,a} \\ k_i \neq k_j \ (i \neq j) \end{array} \right)$$

この最小組合せ数とは、 $N_{1,\min} = 1$ のとき、仕様 S_0 を単独で満足する部分仕様 $S_{0,i}$ があることを示す。次に、この最小組合せ数 $N_{1,\min}$ のとき、式 (3) を満足する $S_{0,i}$ の組合せを抽出し、式 (4) のように、 $C_{1,k}$ で表す。 $K1$ は選択された組合せの総数を表す。

$$\bigcup_{i=1}^{N_{1,\min}} S_{0,k_i} \equiv S_0 \quad (3)$$

$$C_{1,k} = \{S_{0,k_1}, \dots, S_{0,k_{N_{1,\min}}}\} \quad (4)$$

$$(k = 1, \dots, K1)$$

- (3) 前項で抽出された組合せ $C_{1,k}$ を評価し、最適な $C_{1,k}$ を選択する。選択方法は、式 (5) で $C_{1,k}$ に関する入札評価値 $V_{1,k}$ をそれぞれ求め、式 (6) により最小値 $V_{1,\min}$ となる $C_{1,\min}$ を選ぶ。

$$V_{1,k} = \sum_{i=1}^{N_{1,\min}} \left(LA_{k_i} + \alpha \frac{t_{k_i}}{T_a} \right) \quad (5)$$

$$V_{1,\min} = \min_{1 \leq k \leq K1} V_{1,k} \quad (6)$$

{	$N_{1,\min}$	最小組合せ数
	LA_{k_i}	$AC-A_{k_i}$ の平均負荷値
	α	任意の正数
	t_{k_i}	$I-A$ から $AC-A_{k_i}$ への パケット到達時間
	T_a	入札締切り時間

この評価式 (5) の右辺第 1 項 LA_{k_i} は、 $AC-A_{k_i}$ の稼働する計算機の負荷状態を、第 2 項 t_{k_i}/T_a は、 $AC-A_{k_i}$ までの通信状態を評価する。これにより、最も迅速に処理可能な $AC-A_i$ の組合せを選択できる。

STEP1-6 STEP1-5 で選択された最適な組合せ $C_{1,\min}$ に該当する $AC-A_i$ ($i = 1, \dots, N_{1,\min}$) に DB-AP の作成を依頼する (図 9 (1-6))。依頼後、協調過程は第 2 階層の STEP2-1 に移る。なお、 $C_{1,\min}$ に属する部分仕様 $S_{0,i}$ の間には、重複の可能性があるため、重複部分を取り除いた新たな仕様 $C'_{1,\min}$

を作成し、それにより DB-AP 作成を依頼する。重複仕様部分の除去は、入札結果の評価選択方針に従い、次の手順とする。

- (1) 部分仕様の組合せ $C_{1,\min}$ の中から、(a) 仕様 S_0 に対する仕様実現率の大きさ、(b) $AC-A_i$ の処理の迅速性、を考慮し、最良の仕様 $S_{0,i}$ を抽出する。
- (2) 残りの仕様 $S_{0,j}$ ($j \neq i$) について、式 (7) により、抽出した仕様 $S_{0,j}$ との重複仕様を取り除く。

$$S_{0,j} = S_{0,j} - (S_{0,j} \cap S_{0,i}) \quad (7)$$

- (3) 抽出仕様 $S_{0,i}$ を $C'_{1,\min}$ に移動し、 $C_{1,\min}$ から削除する。すべてが $C'_{1,\min}$ に移動されるまで、(1)~(3) の処理を繰り返す。

STEP1-7 第 2 階層での構築処理 STEP2-8 が終了すると、このステップが呼び出される。 $I-A$ は、 $AC-A_i$ へ作成依頼後、一定時間 T_b の間、第 2 階層での構築処理を待つ (図 15 (1-7) 参照)。

- (1) 契約に基づき、すべての $AC-A_i$ から、DB-AP が転送されれば、それらを組み合わせ、ユーザに提供する。このとき、複数の DB-AP は、「統計方法」の項目に従い、それぞれの出力結果を統合するように組み合わせる。
- (2) すべての DB-AP が転送されないか、DB-AP 作成中にエラーが発生した場合、構築を中断し、すべての $ZAC-A_i$ に構築中断を通知する。

(B) 第 2 階層

第 2 階層では、STEP1-6 において $I-A$ から作成依頼された仕様を、 $AC-A$ が $PP-A$ と協調して複数の副仕様に分解する。そして、その副仕様に基づき、部品オブジェクトを作成し、組み合わせさせてアプリケーションを構築する。協調過程は、STEP2-1 から STEP2-8 の 8 ステップより構成される。以下、 $I-A$ から依頼された仕様 $S_{0,i}$ を 1 次仕様 S_1 と呼ぶ。同様に、作成を依頼された $AC-A_i$ を $AC-A$ 、複数の $PP-A$ を $PP-A_i$ ($i = 1, \dots, M_{PP-A}$) と表す。 M_{PP-A} は、 $PP-A$ の総数を表す。

STEP2-1 $AC-A$ は、知識を用いて 1 次仕様 S_1 を展開し、2 次仕様 S_2 を作る (図 12 (2-1))。たとえば、図 6 の知識を持つ $AC-A$ が図 11 の仕様 $S_{0,0}$ を展開した場合、図 13 に示す仕様 S_2 を作る。

STEP2-2 $AC-A$ は 2 次仕様 S_2 をすべての $PP-A_i$ に公示し、入札を求める (図 12 (2-2))。

STEP2-3 提示された $PP-A_i$ は、仕様 S_2 と自己の知識を比較し、実現可能性を判断する。そして、その

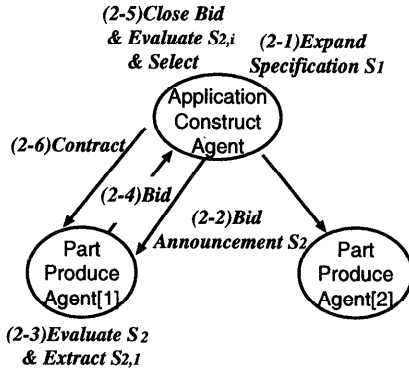


図12 第2階層における協調手順(その1)

Fig. 12 Cooperation procedure in second layer (No.1).

$S_2 = \{$
 検索対象 = {
 Table = (Table: 架空配線, Table: 架空光配線,
 Table: ビル配線) }
 検索条件 = {
 Column: ケーブル方式 = (' 架空')
 Column: 種別コード = (04-10,29,56,79-84)}
 集計項目 = {Column: ケーブル長 }
 集計条件 = {Column: ケーブル建設年
 on (1991~1995)}
 集計方法 = {Function: 総和 } }
 $\}$

図13 2次仕様の例: S_2 Fig. 13 An example of second specification: S_2 .

判断に基づき、部分仕様 $S_{2,i}$ を作成する(図12(2-3)). 実現可能性の判断は、次の3段階で行う。

- (1) 「検索対象」項目に関する知識の有無を確認する。複数のテーブルが指定されている場合、そのうちの1つでも保有すれば、次の判断に進む。同時に、「集計方法」項目で指定されている演算関数に関する知識の存在も確認する。
- (2) 次に、「検索対象」、「集計方法」以外の項目に指定されている全カラムについて、知識中のテーブル情報との一致性を確認する。
- (3) 最後に、各項目中のカラムに対するデータ条件と、知識中のカラムに関するデータ種別情報との整合性を確認する。

以上の判断により、全部、または、一部の仕様を実現可能な場合、 $PP-A_i$ は S_2 から部分仕様 $S_{2,i}$ を抽出する。逆に、作成不可能と判断した場合、すべての項目が空な部分仕様を作成する。たとえば、図8の知識を持つ $PP-A_0$ に、図13の2次仕様 S_2 を提示した場合、図14の部分仕様 $S_{2,0}$ を抽出する。

STEP2-4 $PP-A_i$ は部分仕様 $S_{2,i}$ と $PP-A_i$ が稼働するホストの平均負荷値 LA_i の2項目を用いて、 $AC-A$ に応札(逆提案)する(図12(2-4)).

$S_{2,i} = \{$
 検索対象 = {
 Table = (Table: 架空配線)}
 検索条件 = {
 Column: ケーブル方式 = (' 架空')
 Column: 種別コード = (04-10,29,56,79-84) }
 集計項目 = {Column: ケーブル長 }
 集計条件 = {Column: ケーブル建設年
 on (1991~1995)}
 集計方法 = {Function: 総和 } }
 $\}$

図14 部分仕様の例: $S_{2,0}$ Fig. 14 An example of extracted specification: S_2 .

STEP2-5 $AC-A$ は入札公示から一定時間 T_c 後に入札を締め切る。そして、入札結果 $(S_{2,i}, LA_i)$ ($i = 1, \dots, M_{PP-A,a}$) を評価し、仕様 S_2 を満足する最適な部分仕様 $S_{2,i}$ の組合せを選択する(図12(2-5)). $M_{PP-A,a}$ は、仕様 S_2 に入札してきた $PP-A_i$ の総数を表す。なお、入札結果の評価選択方針、アルゴリズムは、STEP1-5と同様とし、仕様に対する実現能力と構築の迅速性を評価する。

- (1) まず、式(8)により、 S_2 の実現可能性を判断する。成立の場合、次項に進む。逆に、不成立の場合、依頼元の $I-A$ に作成失敗のエラーを返す。

$$\bigcup_{i=1}^{M_{PP-A,a}} S_{2,i} \equiv S_2 \quad (8)$$

- (2) 次に、すべての部分仕様 $S_{2,i}$ の組合せから、 S_2 を満足し、かつ、その組合せ数が最小な組合せ $C_{2,k}$ ($k = 1, \dots, K2$) を抽出する(式(9)). $K2$ は組合せの総数であり、 $N_{2,\min}$ は最小組合せ数である。

$$C_{2,k} = \{S_{2,k_1}, \dots, S_{2,k_{N_{2,\min}}}\} \quad (9)$$

- (3) 最後に、式(10)により、抽出された組合せ $C_{2,k}$ に関する入札評価値 $V_{2,k}$ を求め、式(11)で最小値 $V_{2,\min}$ となる $C_{2,\min}$ を選択する。

$$V_{2,k} = \sum_{i=1}^{N_{2,\min}} \left(LA_{k_i} + \beta \frac{t_{k_i}}{T_c} \right) \quad (10)$$

$$V_{2,\min} = \min_{1 \leq k \leq K2} V_{2,k} \quad (11)$$

$$\left\{ \begin{array}{ll} N_{2,\min} & \text{最小組合せ数} \\ LA_{k_i} & PP-A_{k_i} \text{の平均負荷値} \\ \beta & \text{任意の正数} \\ t_{k_i} & AC-A \text{から} PP-A_{k_i} \text{への} \\ & \text{パケット到達時間} \\ T_c & \text{入札締切り時間} \end{array} \right.$$

STEP2-6 $AC-A$ は、選択された組合せ $C_{2,\min}$ に

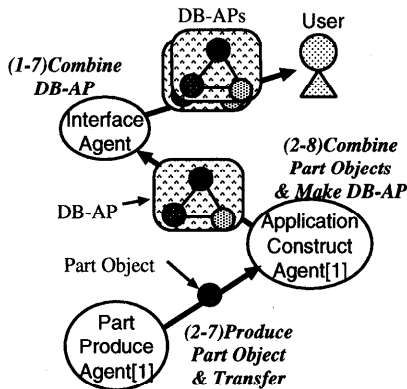


図15 第2階層における協調手法(その2)

Fig. 15 Cooperation procedure in second layer (No.2).

```

select sum(架空配線. ケーブル長) from 架空配線
where (架空配線. ケーブル方式 LIKE '架空'
AND (架空配線. 種別コード LIKE '04'
OR 架空配線. 種別コード LIKE '05'
OR 架空配線. 種別コード LIKE '06' ...))
AND (架空配線. ケーブル建設年 >= 1981
AND 架空配線. ケーブル建設年 <= 1986))

```

図16 部品オブジェクトに埋め込まれたSQLスクリプトの一部
Fig. 16 A part of SQL script included in a part object.

該当する $PP-A_i$ ($i = 1, \dots, N_{\min}$) と契約し、部品オブジェクト作成を依頼する(図12(2-6)). なお、各部分仕様間に重複の可能性があるため、STEP1-6と同様に、重複を取り除き、作成を依頼する。

STEP2-7 作成を依頼された $PP-A_i$ は、知識・機能を用いて、仕様に適合した検索・統計処理機能を持つ部品オブジェクトを作成する(図15(2-7)). たとえば、図14に示す部分仕様 $S_{2.0}$ に対する部品オブジェクトは、図16に示すSQLスクリプトを保持し、このスクリプトに応じた処理を行う。生成後、部品オブジェクトを $AC-A$ に転送する。

STEP2-8 $AC-A$ は $PP-A_i$ へ作成依頼後、一定時間 T_d の間、部品オブジェクトが転送されるのを待つ(図15(2-8)).

(1) すべての部品オブジェクトが転送された場合、それを組み合わせ、1つのDB-APとし、依頼元の $I-A$ に転送する。以降の処理は第1階層に戻る(→STEP1-7)。なお、部品オブジェクトはそれぞれの処理結果を集計するように組み合わせる。たとえば、図14に示す $S_{2.0}$ に相当する部品オブジェクトの場合、その出力は表4のようになり、かつ、仕様の「集計方法」が「総和」であるため、「ケーブル建設年」=「1991」等の各レコードごとに、集計

表4 部品オブジェクトの出力例

Table 4 An example output format of a part-object.

ケーブル建設年	ケーブル長の総和
1991	1500.0
...	...
1995	1235.0

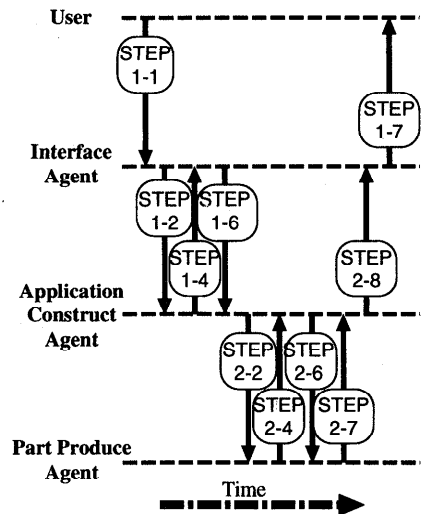


図17 エージェント間メッセージフロー

Fig. 17 Message flow between a user and agents.

値を加算するように組み合わせる。

(2) 部品オブジェクトが転送されないか、 $PP-A_i$ でエラーが発生した場合には、構築を中断し、 $I-A$ にエラーを返す。同時に、すべての $PP-A_i$ に構築中止を通知する。

(C) 協調過程全体のエージェント間メッセージフロー以上、(A)、(B)に示した2階層にわたる協調過程により、複数のエージェントが協調し、DB-APを構築する。ここでは、その協調過程におけるユーザと3種類のエージェントの間で取り交わされるメッセージについて、各STEPと対応させ、図17に示す。なお、横軸が時間軸である。各ベクトルが1つのメッセージを表し、その始点・終点がメッセージの発信者・受信者を表す。

4. 実装と評価

3章で述べた構築手法をプロトタイプシステム(図18)として実装し、その動作を確認した。

4.1 実装環境

実装環境と各エージェントの配置状況を図19に示す。Host1~4の各計算機(CPU: Pentium 133 MHz)は、ルータを介した2つのネットワーク(速度10 Mbit/sec)に接続されている。そして、この環境上



図 18 OPENSTEP 上に実装したプロトタイプシステム
Fig. 18 Prototype system on OPENSTEP.

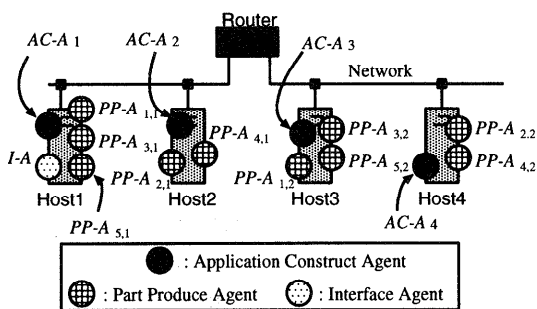


図 19 プロトタイプのネットワーク構成とエージェントの配置状況
Fig. 19 Network structure of prototype system and state of agents on hosts.

に OPENSTEP¹⁰⁾の分散オブジェクト技術を用いて実装した3種類のエージェント $I-A$, $AC-A_i$, $PP-A_{j,k}$ を分散配置した。なお, $AC-A_i$, $PP-A_{j,k}$ にはそれぞれ次の知識を持たせた。

- $AC-A_i$ 「架空ケーブル」($i = 1, 2$) と 「地下ケーブル」($i = 3, 4$) に関する知識
- $PP-A_{j,1}$, $PP-A_{j,2}$ 「架空配線」($j = 1$) 「架空光配線」($j = 2$) 「ビル配線」($j = 3$) 「地下配線」($j = 4$) 「ビル地下配線」($j = 5$) の各データベーステーブルに関する知識

4.2 動作試験

このシステムに作成仕様を入力したときの動作結果を示す。仕様は、図 10 の S_0 とし、これは実システムで利用している DB-AP と同一のものである。さらに、エージェントが保持する知識で作成可能である。なお、入札評価式 (5), (10) の定数 α , β はそれぞれ 300.0 とし、各時間定数 $T_a = 10.0$ [sec], $T_b = 180.0$ [sec], $T_c = 15.0$ [sec], $T_d = 120.0$ とする。

ユーザが仕様 S_0 を $I-A$ に入力したときの構築過程を示す。

- (1) $I-A$ が $AC-A_i$ に仕様を提示すると、各 $AC-A_i$ は部分仕様 $S_{0,i}$ を作成し、応札 (逆提案) した。応札を待って、 $I-A$ は最小組合せ数 $N_{\min} = 2$ と、部分仕様の組合せ $C_{1,l}$ ($l = 1, \dots, 4$) を抽出し、それぞれを評価した。そして、 $C_{1,4} = (S_{0,2}, S_{0,4})$ を選択し、該当する $AC-A_2$, $AC-A_4$ に DB-AP の作成を依頼した。
- (2) 依頼された $AC-A_i$ は 2 次仕様 S_2 を作成し、すべての $PP-A_{j,k}$ に提示した。これに対し、各 $PP-A_{j,k}$ は応札 (逆提案) した。すると、 $AC-A_i$ は入札値を評価し、次の組合せを選択した。

$$AC-A_2 : PP-A_{1,1}, PP-A_{2,1}, PP-A_{3,1}$$

$$AC-A_4 : PP-A_{4,2}, PP-A_{5,2}$$

続けて、 $AC-A_i$ は選択した $PP-A_{j,k}$ に部品オブジェクト作成を依頼し、作成された部品を組み合わせて、DB-AP を構築した。最後に、 $I-A$ がそれらをまとめ、ユーザの仕様を満たす DB-AP を作成できた。

4.3 動作の評価

以上、本構築手法の動作結果を示した。最終的に図 1 に示すような、実際の社内システムで使用可能な DB-AP を作成できた。構築完了には、約 3 分を要した。一方、この仕様を文献 3) に示す従来のオブジェクト指向開発環境を用いて作成した場合、どんなに熟練したユーザでも、構築に 30 分を要した。この両手法の比較から、本構築手法はデータベースに関する専門知識が不要なうえに、従来技術よりも、10 倍以上の作成効率の改善を実現できることが推測される。

5. 考察

本構築手法の有効性について考察する。有効性は、DB-AP を利用している実システムに、本構築手法を適用した場合の効果により判断する。というのは、本手法では、ユーザが複雑な仕様を入力し、それに該当するすべての知識をエージェントに与えれば、たいいてい DB-AP を構築することが可能である。しかしながら、現実が必要とされるすべての DB-AP に、本手法を適用することは必ずしも適当ではない。なぜならば、ユーザが複雑な仕様を入力するためには、多くの専門知識が必要となり、本論文の本来の目的に反する。さらに、非常に特殊な構造や領域の場合、従来方法による構築の方が効率的である。以上のことから、本構築手法の適用効果について考察する。

対象とする実システムは、3 章で対象とした、NTT

表5 既存システムに対する本構築手法の適用の有効性
Table 5 Effectiveness of applying our construction method to an actual system.

評価	有効	無効
個数(割合)	75 (51%)	70 (49%)

の加入者系設備管理システムとする。ここで利用されているDB-APは、文献3)に示すオブジェクト指向技術を用いて、ソフトウェアの専門家ではなく、システムを利用するユーザが作成したものである。現時点で、そのDB-APの総数は規模の大小合わせて145種類ある。このすべてについて、本手法の適用の有効性を評価する。有効、無効の判断は次の定性的基準による。
適当 検索条件や集計条件等の各仕様項目において、その構造が類似なDB-APが複数存在し、本手法の導入により効果が予想できる。

不適当 仕様が非常に特殊であり、同じ構造の仕様が複数存在しないか、必要でない。

評価結果を表5に示す。これにより、既存システムの約半分について適用することが有効であり、保守性・拡張性などを改善できることを示している。さらに、これら145種のDB-APはすべて業務経験者が作成したものであり、データベースの知識を得るために長時間の訓練を必要としたものばかりである。その意味で、その半数をプログラミング知識、データベース知識を必要とせずに、作成できる効果は非常に大きい。

6. まとめ

本論文では、従来のオブジェクト指向ソフトウェア構築技術の課題を解決する手法として、分散エージェントの協調による問題解決技術を適用したソフトウェア構築手法を提案した。そして、この構築手法を検証するため、データベースの検索・統計処理を行うデータベースアプリケーションの構築という問題領域に適用し、エージェントの機能、知識表現方法、分散エージェントの協調方法等を具体的に示した。さらに、そのエージェント間協調手法として、従来の契約ネットプロトコルを改良し、入札過程に逆提案を用いることにより、複雑な調停作業を簡潔にした。また、現実の社内システムで利用可能なデータベースアプリケーションを構築するプロトタイプシステムを作成し、動作の検証と、従来方法との比較による評価を行い、実問題に対する有効性を考察した。

最後に今後の検討課題をまとめる。

(1) 本論文では、エージェントを3種類、協調過程を2段階とした。これは、既存システムで利用されているDB-APの構築過程を分析した結果に基づき、

決定した。しかしながら、本構築手法を適用可能な問題領域は、DB-APの構築に限定されるわけではなく、より一般的な構築対象にも適用可能である。ところが、そのような場合、この2段階にわたる協調過程は必ずしも十分ではない。というのは、アプリケーションの構造上の複雑さは、一定とは限らないからである。したがって、今後はより多段階にわたる仕様分割の協調手法を検討する必要がある。しかしながら、この問題については、エージェント間の協調手法に、再帰的な協調過程を導入することにより、多段階にわたる仕様の分割を実現可能である。

(2) 本論文の構築手法では、1つの仕様に対して複数のエージェントから入札があった場合、(a)入札してきたエージェントの仕様実現能力と、(b)構築の迅速性、の2項目を評価し、適当なエージェントを選択する。ところが、今後、構築環境を広域ネットワーク上に展開した場合、非常に類似したエージェントが多数導入され、似て非なるソフトウェア部品が生成される可能性がある。このような場合、入札に対する評価項目は、その部品の機能性や信頼性など、より本質的に評価する必要がある。しかしながら、問題領域をデータベースアプリケーションの構築に限定した場合、生成されるソフトウェア部品の品質は、内部で実行されるSQLスクリプトの品質に対応させることができる。したがって、このような各領域の特徴を考慮した評価式を導入することにより、対処可能である。

謝辞 本研究に対して、有益なコメントをいただいたNTT研究開発本部の桑原課長、NTTソフトウェアの青野課長、NTTアクセス網研究所の薮田プロジェクトマネージャに感謝いたします。また、日頃から様々な意見、コメントをいただいたNTTアクセス網研究所の皆様にも感謝します。

参考文献

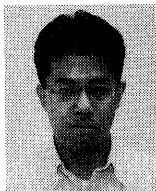
- 1) 長崎 祥, 田中 譲: シンセティックメディアシステム IntelligentPad における協調作業場の実現, 情報処理学会論文誌, Vol.34, No.5, pp.1177-1187 (1993).
- 2) 本位田真一, 山城明宏: オブジェクト指向分析・設計, 情報処理, Vol.35, No.5, pp.392-401 (1994).
- 3) 片岡康郎, 加藤康之, 阿部徹治, 満永 豊: オブジェクト指向による開放型データベースアプリケーション構築支援システムの実現, 電子情報通信学会技術研究報告, KBSE95-23, pp.9-16 (1995).
- 4) Inoue, A., Abe, T., Kato, Y. and Mitsunaga, Y.: Implementation of Software Construction Environment using a Multiagent, *Proc. 2nd Intl. Conf. on Multi-Agent Systems (ICMAS-*

96), p.442 (1996).

- 5) 井上 晃, 阿部徹治, 満永 豊: マルチエージェントを用いたソフトウェア構築環境の実現, 情報処理学会オブジェクト指向'96 シンポジウム, pp.127-134 (1996).
- 6) Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Trans. Comput.*, Vol.29, No.12, pp.1104-1113 (1980).
- 7) 石田 亨, 桑原和宏: 分散人工知能 (1): 協調問題解決, 人工知能学会誌, Vol.7, No.6, pp.945-954 (1992).
- 8) 桑原和宏, 石田 亨: 分散人工知能 (2): 交渉と均衡化, 人工知能学会誌, Vol.8, No.1, pp.17-25 (1993).
- 9) 桑原和宏, 篠原拓嗣, 大里延康, 石田 亨: 協調プロトコル記述言語 AgentTalk の実現, 電子情報通信学会技術研究報告, AI95-18, pp.33-40 (1995).
- 10) Next Software Inc.: OpenStep Specification (1994).

(平成 9 年 9 月 5 日受付)

(平成 10 年 5 月 8 日採録)



井上 晃 (正会員)

平成 6 年大阪大学大学院工学研究科電子制御機械工学専攻修士課程修了。同年日本電信電話(株)アクセス網研究所に入社。現在, 知的ソフトウェア, 分散オブジェクトに関する研究に従事。電子情報通信学会, 日本ソフトウェア科学会各会員。



阿部 徹治 (正会員)

昭和 57 年京都大学大学院工学研究科電気工学専攻修士課程修了。同年日本電信電話公社茨城電気通信研究所入所。以来, 光ファイバの信頼性, 加入者系のオペレーションシステム, オブジェクト指向ソフトウェアの研究に従事。現在, (株)サイバー・ラボ研究開発本部長。



満永 豊 (正会員)

昭和 52 年東京工業大学大学院工学研究科生産機械工学専攻修士課程修了。同年日本電信電話公社茨城通信研究所入所。以来, 光ファイバケーブル構造設計, ソフトウェア開発環境に関する研究に従事。工学博士。



加藤 康之 (正会員)

昭和 52 年山形大学電子工学科卒業。同年日本電信電話公社茨城電気通信研究所入所。以来, 昭和 52~62 年単一モード光ファイバの伝送特性および接続技術の研究, 昭和 63~平成 2 年 NTT 岩手支社企画室長, 平成 3~10 年知的ソフトウェア工学の研究に従事。現在, (株)サイバー・ラボ代表取締役社長。昭和 57 年度電子通信学会論文賞, 昭和 58 年度電子通信学会学術奨励賞。工学博士。